Robust Feature Detection, Acquisition and Tracking for Relative Navigation in Space with a Known Target

Dae-Min Cho* Panagiotis Tsiotras† Guangcong Zhang‡ and Marcus J. Holzinger§
Georgia Institute of Technology, Atlanta, GA, 30332-0150, USA

Recent advances in robotics and computer vision have enabled the implementation of sophisticated vision-based relative navigation algorithms for robotic spacecraft using a single calibrated monocular camera. These techniques, initially developed for ground robots, show great promise for robotic spacecraft applications. However, several challenges still exist, which hinder the direct use of these approaches in the space environment without further modifications. For example, the use of a monocular camera for robotic spacecraft operations with respect to a known target configuration may be limited owing to the abrupt illumination changes in a low-Earth orbit, long duration target tracking requirements during large target image change in scale, background outliers, and the necessity to perform (semi)autonomous relative navigation in the presence of limited resources (fuel, onboard computer hardware, etc). This paper proposes a relative navigation scheme in space that makes use of three different ingredients. First, two different feature detectors are used to ensure reliable feature detection over diverse distances, and subsequently fast feature selection/filtering is applied to detect the visual features of the fiducial marker. Next, a feature-pattern matching algorithm via robust affine registration is used for relative navigation to achieve robust automated re-acquisition in case of a lost target. Finally, a probabilistic graphical model-based fixed-lag smoothing based on factor graphs is used to accurately propagate relative translation and orientation 6-DOF state estimates and their velocities. The proposed approach is validated on hardware-in-the-loop 5-DOF spacecraft simulation facility at Georgia Tech.

I. Introduction and Background

High accurate and robust automated relative navigation is critical for autonomous space rendezvous and proximity operations in space. It is also a crucial ingredient for persistent Space Situational Awareness (SSA) applications. Several relative navigation techniques have been proposed in the literature, and have been tested and utilized, for example: ground station aided relative navigation, GPS in conjunction with IMUs, Lidar-based, etc [1–5]. However, these techniques have various limitations, such as coarse accuracy, multi-path issues, signal loss in case of occlusion by other spacecraft, IMU drift, limited sensor range, etc. Most recently, vision-based estimation by tracking salient features across images has gradually become an attractive alternative for relative space navigation. As vision sensors become smaller and cheaper (and along with their low power consumption), they permit a feasible approach for vision-based relative navigation, especially for small satellites involved in on-orbit servicing, visual inspection of space objects, and similar proximity operations. On the other hand, vision-based techniques for space applications can suffer from limited visibility, loss of resolution at a certain distance from a target, no useful color/contrast information, potential loss of tracking when the target is outside the camera field of view, etc. Furthermore, their performance depends strongly on illumination conditions, which may change very rapidly in space.

For small satellites with limited onboard fuel and poor actuators and sensors, a waypoint strategy (more generally, a glideslope guidance law [6]) can be used for rendezvous in space. In this approach, a sequence

*PhD Candidate, School of Aerospace Engineering, Email:dcho3@gatech.edu, AIAA Student Member.
†Dean's Professor, School of Aerospace Engineering, Email:tsiotras@gatech.edu, AIAA Fellow.
‡PhD Candidate, School of Electrical and Computer Engineering, Email:zhanggc@gatech.edu.
§Assistant Professor, School of Aerospace Engineering, Email:holzinger@gatech.edu, AIAA Senior Member.

of intermediate goals (waypoints) is defined in order to reach the target and mitigate the operational risks stemming from growing estimation errors. The limited amount of on-board fuel restricts attitude control mostly between waypoints, thus resulting in loss of relative navigation accuracy when a vision sensor is used [7]. Such space-specific maneuvers require robust and automated re-acquisition capability to re-initialize the relative navigation process when the target comes back to the camera field of view.

In this paper we present a complete methodology for automated relative navigation using a monocular camera and a known target configuration. A monocular camera setup can arguably increase the estimation accuracy when it comes to a known target, as the calibration effort for a monocular camera is much less than for a binocular (stereo) camera setup. In addition, the calibration error has a lesser effect on state estimation accuracy [8], while a known target configuration does not generate scale ambiguity, which is an issue for the case of an unknown target configuration. Finally, a binocular camera may have a limited operating field of view, which is defined by the overlapped field of view of each camera, as it depends on both the baseline distance between the two cameras and the field of view of each camera.

In this paper, we propose three ingredients for relative pose, and linear and angular velocity estimation for a spacecraft in orbit:

- We use two different feature detectors, namely, the Shi-Tomasi corner detector [9] and the rotation and scale invariant SURF (Speeded Up Robust Features) blob detector [10], followed by a series of feature selection and filtering steps, in order to achieve robust and efficient feature detection performance over a diverse scale of distances to the target. This is important, since the imaged fiducial marker scale changes according to the distance to the target.
- We use a robust feature-pattern matching technique based on Gaussian Mixture Model (GMM) point-set registration for target re-acquisition in order to provide fault-tolerant and automated relative navigation capability.
- We use a fixed-lag smoothing SLAM algorithm based on a probabilistic factor graph model to obtain optimized pose, linear and angular velocity estimates with high accuracy.

The use of SLAM-based estimation methodologies for vision-based relative navigation in orbit, in particular, opens the possibility for injecting further perception capabilities to the spacecraft, thus increasing their autonomy. For some recent related work, see also [11, 12].

The rest of the paper is organized as follows: Section II summarizes related previous work, while Section III provides an overview of the proposed approach. Section IV describes each relative pose estimation step in detail, while Section V presents the probabilistic graphical model-based smoothing method used in this paper to obtain optimized relative pose, and linear and angular velocity estimates. The experimental setup and the results from a series of experiments are presented in Section VI. Finally, we conclude the paper with some possible topics for further research.

II. Related Work

Spacecraft rendezvous and formation flight navigation architectures using optical sensors can be categorized based on the number of cameras used in the implementation. When two cameras are used, the system is called a Binocular (or stereo) architecture, while a single camera architecture is called Monocular. Binocular (stereo) cameras are a popular choice for relative navigation problems since they can directly provide 3D feature points on a target using geometry and a known baseline between the cameras [13,14]. In cooperative formation flight scenarios, detected feature points are associated with points on a known rigid-body model of the target, and the relative orientation & position of the target is estimated. The relative position accuracy of 3D features found using binocular vision approaches degrades as the distance to features increases. This degradation results from the reduced feature disparity between each camera, which is inversely proportional to target distance when the baseline distance is fixed. Monocular cameras, however, require fewer system resources (space/weight/power) and less calibration effort than binocular camera systems. Further, monocular camera systems are able to cope with the fixed baseline constraint of the binocular cameras by employing adaptively large enough transversal relative displacement of the chaser required for triangulation. Recent

work has shown the feasibility of relative spacecraft navigation using monocular cameras for fully cooperative cases using Extended Kalman Filters [15].

Vision-based pose determination usually requires fast feature point matching between consecutive images. Point matching between consecutive images is enabled by several point matching methods, mainly using descriptor vectors [10, 16]. However, the space environment hinders the use of descriptor-based point matching methods owing to the harsh illumination conditions in space (the descriptor vector is encoded by the values of color or the gradient around a detected feature point). To address this problem, feature-pattern matching via point-set registration has been used in the literature. Point-set registration algorithms fall into two main classes: parametric model-based and physical model-based methods. Parametric model-based algorithms approximate the deformation fields using bases functions, and subsequently optimize over the approximated representation. Typical bases functions include Gaussian basis functions [17–19] and thin-plate splines [20]. On the other hand, physical model-based methods use physical transformation models, e.g., an affine transformation [21] to represent the deformation and to register point-sets by recovering the physical transformation model.

Relative pose estimation of a calibrated camera from known correspondences of 3D feature points in the target frame and their associated 2D image points in the camera frame is called the *exterior orientation problem* or the *space resection problem* in photogrammetry. The Perspective-n-Point (PnP) problem refers to the pose estimation problem from n 2D-3D correspondence points. Among the various PnP methods, P3P has been extensively studied in the literature [22–25]. It is well known that P3P has a closed form solution [23], but yields up to four possible solutions. If a fourth point is added on the same plane formed by the three other points, the coplanar P4P problem yields a unique solution [23]. It is also known that non-coplanar P6P yields a unique solution as well. Recently, Lepetit at el [26] introduced the Efficient PnP (EPnP) algorithm, whose computational complexity grows only linearly with the number of correspondence points. The EPnP algorithm provides an accurate closed-form solution efficiently; the solution can be further refined using the closed-form solution to initialize an iterative method (such as Gauss-Newton or Levenberg-Marquardt) to solve a nonlinear optimization problem. However, experimental results have shown that the closed-form solution of the coplanar EP4P problem is accurate enough so there is no need for solving the nonlinear optimization problem.

When a measurement is established by solving the point matching and correspondence problems, the measurement is typically filtered in order to estimate the states with better accuracy than just using raw sensor measurements. The EKF (Extended Kalman Filter) solution is a popular choice when the system is mildly nonlinear. The EKF uses a history of measurements to build a model of the state that minimizes the trace of the error covariance of the state. It is a recursive state estimation algorithm that is numerically efficient. This are other various recursive filtering techniques such as particle filters (PF) have been applied to the problem of vision-based relative navigation for space applications [7, 15]. They have been shown to provide a fast estimate of the most recent state in constant time [27]. Smoothing is another framework for state estimation, which is known to provide a better accuracy solution [27] (but not in constant time). Smoothing works by keeping a history of the states and by correcting them to account for wrong linearization choices. Recently, both filtering and smoothing frameworks for the navigation problem have been cast using probabilistic graphical models [28]. The latter approach is based on simple topological analysis [29–31] and may result in increased accuracy using fused sensor measurements [27, 32].

III. Overview of Proposed Approach

The proposed algorithm assumes that the system takes a sequence of images of the target as an input, and outputs the estimated relative pose of the chaser with respect to the target spacecraft. It also assumes a fully calibrated monocular camera. The algorithm incorporates three key ideas. First, in order to cope with a wide range of target detection distances, two different feature point detectors are employed: the Shi-Tomasi corner detector, and the SURF blob detector. Blob (Binary Large OBject) detectors with a monocular camera have been widely used in space applications to find simple contours along with their centroids on a target [7,15]. In our setup, these two feature point detectors are automatically engaged, based on the distance to the target and the availability of trackable features. At initialization, the point pattern is matched to the known target configuration using a robust feature-pattern matching based on GMM point-set registration (for details of the the feature matching algorithm, see Section IV.B). Finally, a graphical

model-based fixed-lag smoothing is employed to increase estimation accuracy.

The specific steps of the approach are listed below:

- Noise reduction: A Gaussian filter is applied to the image to reduce high-frequency spatial noise.
- Feature detection and selection: The Shi-Tomasi detector and the SURF detector are used to obtain a raw feature set. The features from the markers are selected by a feature selection/filtering step based on neighborhood information, local color information and motion coherence.
- Feature matching: The a priori known feature point pattern is matched to the target using a robust feature-pattern matching technique based on GMM point-set registration. A 4-point algorithm (EP4P) is subsequently applied to get the initial pose of the chaser with respect to the target.
- Feature tracking: The Kanade-Lucas-Tomasi (KLT) tracker is used to track feature points from one image to the next. This avoids the use of direct feature detection for every image, as in that case feature tracking will not be stable because previously detected features may not be detected again due to noise, different illumination conditions, etc.
- Pose determination: The pose of the chaser is computed based on the tracked feature points and their matching with the known target. The EP4P algorithm is used in this paper since it provides good accuracy at acceptable computational speed, without requiring iterative refinement.
- Main loop: Go to feature tracker, pose determination, relative state estimation, and repeat; periodically also run the feature detector to update the new feature points since the previously tracked feature points drift with time due to noise.

The schematic of the overall algorithm is given in Fig. 1.

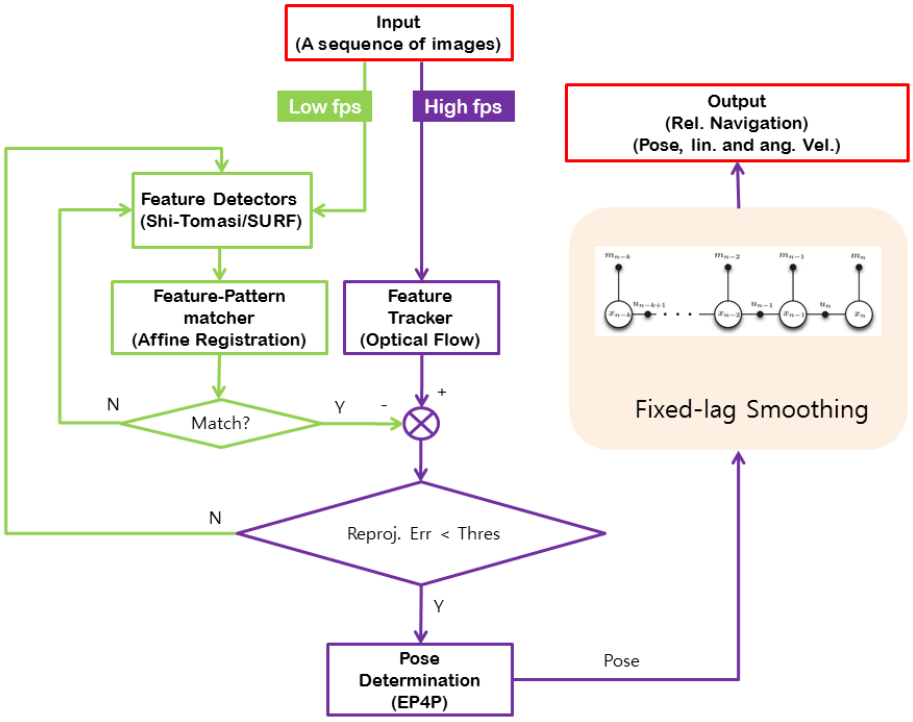


Figure 1. Overall schematic of proposed approach.

IV. Relative Pose Estimation

IV.A. Feature Detection

In computer vision, "features" denote any "meaningful" or "interesting" image structure elements such as corners, edges, curves, colors, regions, to be used for object detection. Good features should be informative, insensitive to noise and illumination conditions, and they should be invariant to geometric transformations such as translation, rotation, and scale. Space applications, in particular, may be characterized by periodic illumination changes, long operational range, and cluttered background, and these are often the main criteria for selecting a proper feature detector.

The fact that a centroid of a circular blob is invariant to translation, rotation, and scale change in an image, makes them a common choice for use as feature target points [15]; indeed, circular fiducial markers are often used as target points in space visual navigation experiments. Infrared LED emitters are also often used instead of circular fiducial markers in order to remedy the changing illumination conditions in space. In real applications, a circular blob will be seen as a small dot at some distance from the target. In such cases, blob analysis (i.e., finding contours and/or the centroid) will not work. To cope with such a scenario, in this work it is proposed to use two different feature detectors, depending on the availability of the feature points at different ranges. At close range, a scale invariant blob detector (SURF) is applied to find the centroids of the circular fiducial markers, while at far range, a corner detector (Shi-Tomasi) is applied since the circular fiducial markers are too small to apply multi-scale operations as the SURF does.

The Shi-Tomasi feature detector [9] finds good features to track based on the corners detected by a Harris corner detector. The Harris corner detector [33] is an eigenvalue-based feature point detector and is the most widely used corner detector. It finds a corner feature by taking an image patch around a point of interest and by calculating the intensity gradient up to first order Taylor series in order to find the autocorrelation matrix (M). The autocorrelation matrix provides a measure of corner quality (also called corner response) such that two large eigenvalues of the matrix represents a corner. Only values higher than a given threshold are regarded as corners. On the downside, the Harris corner detector usually finds many redundant corners, as the corner response is calculated locally in the image, and in practice, non-maximum suppression is applied to find a local maximum in 3x3 neighborhood. The Shi-Tomasi's algorithm selects those features that can be tracked well using the minimum of the two eigenvalues as a corner response measure for image patches undergone affine transformations. Corner responses bigger than a predefined threshold are selected as a corner and expressed as

$$\min(\lambda_1, \lambda_2) > \varepsilon, \tag{1}$$

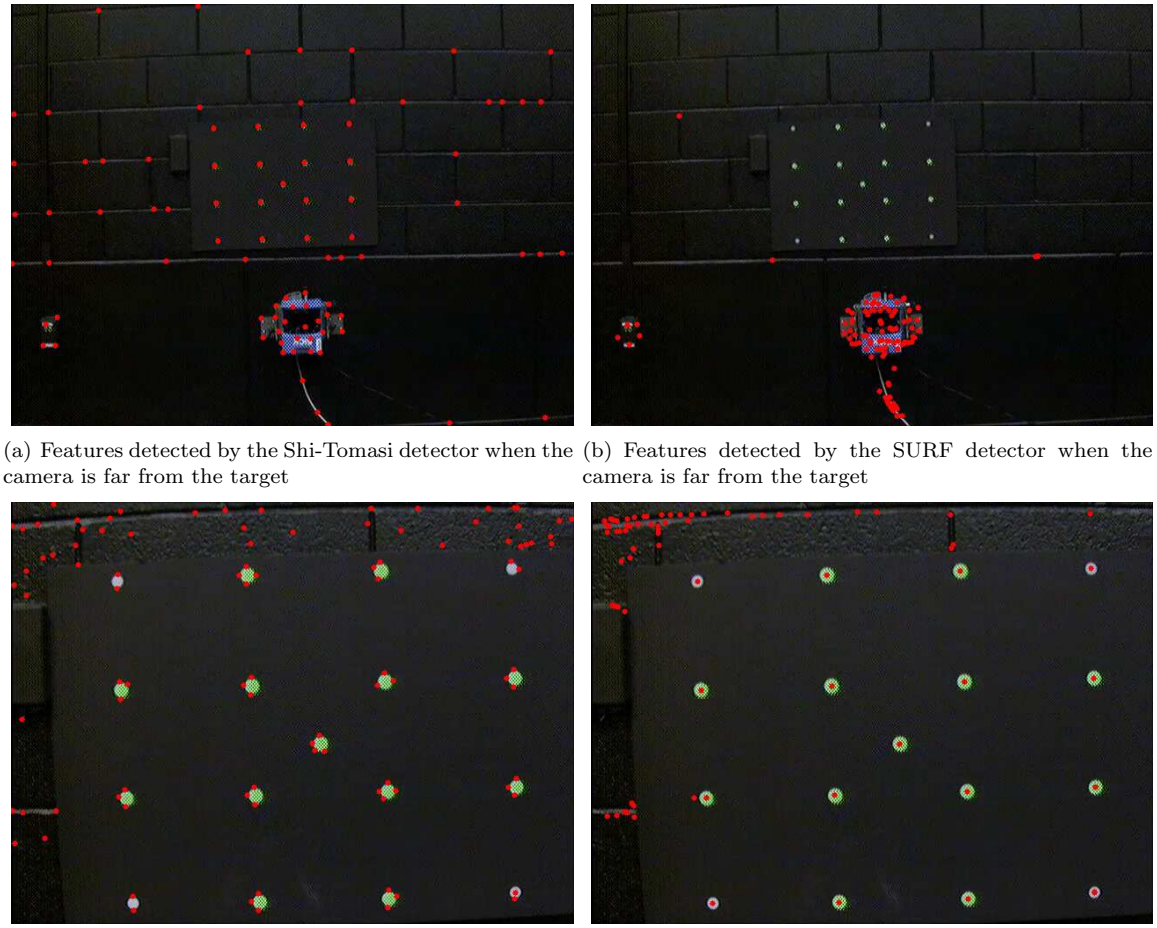
where λ_1 and λ_2 are the eigenvalues of the autocorrelation matrix of the image, and ε is a threshold. Usually, the corner locations are further refined to a sub-pixel accuracy by error minimization or other iterative methods.

SURF (Speeded Up Robust Features) [10] is a scale space-based rotation and scale-invariant feature point detector and descriptor. Descriptors encode an image patch around a detected feature point to match the same feature point detected on the other image. The encoding is done using color, texture, intensity gradient, etc. As a result, it is sensitive to illumination conditions and thus is not readily suitable for space applications. Thus, it was decided that a point pattern matching algorithm to be used instead. Hence, only the detector part of the SURF is explained here. Out of several scale space-based detectors/descriptors, SURF has been adopted due to its good detection speed and repeatability. In this context, repeatability means that the feature detector can robustly find the same physical point in an image taken from different view points. A convolution of a Gaussian filter with the image is usually used for smoothing, then sub-sampling of the image gives a higher level of the image pyramid (a discrete representation of a sampled image in both space and scale) in the scale space, but SURF increased its speed by up-scaling the filter instead of sub-sampling the image. The determinant of the Hessian matrix of the convolution is used as a blob response measure, and the local maxima in scale and image space are selected as feature points. The computational benefit of the SURF algorithm comes from approximating the Hessian matrix with box filters in conjunction with integral images.

IV.B. Feature Matching

Feature Detection Scheme

A small dot has large corner response when a Shi-Tomasi detector is used but this is not so for a SURF detector, as the dot is too small to scale up in scale space. The opposite holds true for large dots. Thus, both feature detectors are used to obtain the original set of features. This set is further processed by the feature selection and filtering steps. Figure 2 illustrates the feature detection resulting from the Shi-Tomasi and SURF detectors, when the camera is far from, and close to the target, respectively. As shown in Fig. 2, when the camera is far from the target, the Shi-Tomasi detector gives better detection results of the markers, while at close range, the SURF detector detects the features centered at the marker circles more accurately.



(a) Features detected by the Shi-Tomasi detector when the camera is far from the target

(b) Features detected by the SURF detector when the camera is far from the target

(c) Features detected by the Shi-Tomasi detector when the camera is close to the target

(d) Features detected by the SURF detector when the camera is close to the target

Figure 2. Features detected by Shi-Tomasi detector/ SURF detector when the camera is far from/ close to the target.

Fast Selection and Filtering of Detected Features

After the feature detection step, visual features from the whole frame are captured if they satisfy the detection criteria of the detector. However, since the detected features may be redundant, e.g., more than one feature is detected from a single marker dot, filtering needs to be employed to filter out the redundant neighboring features. Moreover, because the detected features may not lie on the fiducial marker target, and in order to make the later step of feature pattern matching more robust, further steps need to be applied to select the features from the marker. Therefore, after the feature detection step, selection and filtering are applied on the features based on the neighborhood clustering, local color information and frame-to-frame motion coherence.

The feature selection and filtering procedure starts with the neighborhood clustering on all N detected features to reduce redundancy. For each detected feature $x_i \in \{x_i\}_{i=1}^{N}$ with image position $[a_{x_i}, b_{x_i}] \in \mathbb{R}^2$, all the features $\{x_j\} \in \mathcal{N}(x_i)$ in a local neighborhood of x_i are extracted with the image positions $\{[a_{x_j}, b_{x_j}]\}$. Noted that x_i also lies in $\mathcal{N}(x_i)$, i.e., $x_i \in \{x_j\}$. Then for each feature $x_j \in \{x_j\}$, move x_j to the centroid position $[a_{\mathcal{N}(x_i)}, b_{\mathcal{N}(x_i)}] = [\overline{a_{x_j}}, \overline{b_{x_j}}]$. After placing each feature with the neighborhood centroid, a unique scan operation is conducted to keep only the unique centroids, which are used as the updated feature set $x_i \in \{x_i\}_{i=1}^{N'}$. Since the centroid replacing operation of each feature is of a linear runtime, the overall complexity for N features is $\Theta(N)$.

After neighborhood clustering, a simple but effective selection step is applied with the color information of the local neighborhood block of each feature. This selection step is based on the observation that the color of the fiducial marker dots is composed by a mixture of green and blue colors with certain ratio (while the red-channel value may vary based on the illumination). The selection consists of three sub-steps. Firstly a local neighborhood block $\mathbf{B}_{x_i}$ of each feature $x_i \in \{x_i\}_{i=1}^{N'}$ is extracted. Then the trimmed-mean $\overline{G_g}$ of the green-channel is computed with the K highest elements of the green-channels in $\mathbf{B}_{x_i}$. Similarly, trimmed-mean $\overline{G_b}$ is also computed for the blue-channel. The trimmed-mean is used to make the selection more robust since the feature may not be located exactly at the center of the marker dot. Finally, the feature is selected if the ratio $\frac{\overline{G_g}}{\overline{G_b}}$ is within $\tau \pm \delta$. The ratio value is used instead of the absolute values in order to make the selection invariant to illumination. Denote the updated feature set as $x_i \in \{x_i\}_{i=1}^{N''}$. Overall, this selection step is also of linear runtime, i.e. $\Theta(N')$.

Finally, the motion coherence is checked for each feature $x_i \in \{x_i\}_{i=1}^{N''}$ by only keeping the features located within the motion coherence bounded box $\mathbf{B}_{Motion}$. $\mathbf{B}_{Motion}$ is computed from the final selected feature set $[a'_i, b'_i] = x'_i \in \{x'_i\}$ of the previous frame by defining

$$\mathbf{B}_{Motion} \triangleq \{[a,b] \mid a \in [\min(\{a'_i\}) - R_M, \max(\{a'_i\}) + R_M], b \in [\min(\{b'_i\}) - R_M, \max(\{b'_i\}) + R_M]\} \quad (2)$$

In the case of initialization, or when the marker target is not detected within the frame, $\mathbf{B}_{Motion}$ is set to be the whole image range. After filtering the features, if the marker is recognized from the current frame, then $\mathbf{B}_{Motion}$ is updated with the final selected features of the current frame.

In summary, because each of the three sub-steps is of linear runtime, the final complexity of the selection/filtering step is of order $O(N)$.

Feature Pattern Matching by Robust Affine Registration using GMM

Having the feature set of the current frame after the selection/filtering step, a point-set matching step is employed by registering the feature point-set in terms of pixel positions $\{\mu_i\}_{i=1}^{m}$ with the pattern of the fiducial marker $\{\nu_j\}_{j=1}^{n}$ under an affine transformation. The point-set registration is performed by minimizing the L_2 distance of the GMMs built with the feature point-set and the pattern point-set [21, 34]. Assuming the point-set as a collection of Dirac Delta functions, then we use a GMM as a continuous approximation of the point-set. The number of the components in the GMM is the cardinality of the point-set and the mean vector is given by the location of each point. Also, in our case each component of the GMM is assumed to be of spherical Gaussian with the same mixture weight. Then the GMM of feature point-set with m features is expressed as:

$$f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i \phi(\mathbf{x}|\mu_i, \Sigma_i) \quad (3)$$

where

$$\phi(\mathbf{x}|\mu_i, \Sigma_i) = \frac{1}{2\pi\sqrt{|\det(\Sigma_i)|}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mu_i)^\top \Sigma_i^{-1}(\mathbf{x}-\mu_i)\right] \quad (4)$$

Under the affine transformation $\{\mathbf{A}; \mathbf{t}\}, \mathbf{A} \in \mathbb{R}^{2\times 2}, \mathbf{t} \in \mathbb{R}^{2\times 1}$ in the Euclidean space, the GMM of the transformed feature point-set is given by:

$$f_{\mathbf{A};\mathbf{t}}(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i \phi(\mathbf{x}|\mathbf{A}\mu_i + \mathbf{t}, \mathbf{Q}\Sigma_i\mathbf{Q}^\top) \triangleq \sum_{i=1}^{m} \alpha_i \phi(\mathbf{x}|\widetilde{\mu}_i, \widetilde{\Sigma}_i) \quad (5)$$

where $\mathbf{Q}$ is the orthogonal matrix from polar decomposition $\mathbf{A} = \mathbf{QS}$.

The point-set of the marker pattern is also modeled with a GMM. Letting $g(\mathbf{x}) = \sum_{j=1}^{n} \beta_j \phi(\mathbf{x}|\nu_j, \Gamma_j)$ be the GMM of the pattern point-set, then the distance between the pattern point-set and the affine transformed point-set can be defined by the L_2 distance between the two GMMs:

$$D(f_{\mathbf{A};\mathbf{t}}, g) \triangleq \int (f_{\mathbf{A};\mathbf{t}} - g)^2 \, \mathrm{d}x = \int (f_{\mathbf{A};\mathbf{t}}^2 - 2 f_{\mathbf{A};\mathbf{t}} g + g^2) \, \mathrm{d}x \tag{6}$$

Notice that for the Gaussian distribution ϕ in Eq.(4), we have

$$\int \phi(\mathbf{x}|\mu_1, \Sigma_1)\phi(\mathbf{x}|\mu_2, \Sigma_2) \, \mathrm{d}x = \phi(0|\mu_1 - \mu_2, \Sigma_1 + \Sigma_2). \tag{7}$$

Therefore, $D(f_{\mathbf{A};\mathbf{t}}, g)$ is computed effectively in closed-form:

$$D(f_{\mathbf{A};\mathbf{t}}, g) = \sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j \phi(0|\widetilde{\mu}_i - \widetilde{\mu}_j, \widetilde{\Sigma}_i + \widetilde{\Sigma}_j) + \sum_{i=1}^{n}\sum_{j=1}^{n} \beta_i \beta_j \phi(0|\nu_i - \nu_j, \Gamma_i + \Gamma_j) - 2\sum_{i=1}^{m}\sum_{j=1}^{n} \alpha_i \beta_j \phi(0|\widetilde{\mu}_i - \nu_j, \widetilde{\Sigma}_i + \Gamma_j) \tag{8}$$

The affine transformation is then recovered by minimizing $D(f_{\mathbf{A};\mathbf{t}}, g)$ with respect to $\mathbf{A}$ and $\mathbf{t}$, that is,

$$\hat{\mathbf{A}}, \hat{\mathbf{t}} = \arg\min_{\mathbf{A},\mathbf{t}} D(f_{\mathbf{A};\mathbf{t}}, g). \tag{9}$$

Since the gradient of $D(f_{\mathbf{A};\mathbf{t}}, g)$ is computed in a closed-form, a gradient-based nonlinear optimization method, for instance, the L-BFGS method, can be used to solve this minimization problem.

To make the registration more robust, both the feature point-set and the pattern point-set are normalized before the affine-transformation recovery. After the optimization in the normalized space, the affine-transformation on the original point-sets is obtained by de-normalization. More specifically, given an input point-set $\{\mu_i\}_{i=1}^{m}$, the relationship between $\{\mu_i\}$ and the normalized point-set $\{\mu_i'\}$ is

$$\mu_i' = K(\mu_i - \overline{\mu_i}), \tag{10}$$

where $K = \sqrt{m}/\|\mu_i - \overline{\mu_i}\|_{\mathrm{F}}$, and $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm of a matrix [35].

The declaration of the feature-pattern matching is made based on the GMMs distance in the normalized space. If the minimized L_2 distance is less than a pre-defined threshold, then we claim that the marker pattern is observed in the current frame.

IV.C. Feature Tracking

After the features have been detected and matched, it is common to track the features, rather than repeatedly perform the feature detection and matching process at the next step, since the feature matching process is computationally expensive. In addition, if feature detection is performed for every image, the previously detected features may not be detected again due to noise, changed illumination conditions, etc, thus resulting in high drift or even instability of the feature tracking algorithm. In computer vision, feature matching (equivalently, solving the so-called correspondence problem) between different frames is still a challenging task; yet, good feature matching is essential for accurate camera pose estimation. Once good feature matching has been established, a feature *tracker* usually works with good accuracy for some time. In this paper, the well-known KLT tracker [36] has been used to track feature points across images in order to perform robust pose estimation.

Feature Tracking Evaluation

A perspective projection matrix $\boldsymbol{P} \in \mathbb{R}^{3\times4}$ is expressed as

$$\boldsymbol{P} = \boldsymbol{K}[\boldsymbol{R} \;\; \boldsymbol{t}], \tag{11}$$

where $\boldsymbol{K} \in \mathbb{R}^{3\times3}$ is the intrinsic camera matrix obtained by camera calibration, and $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ and $\boldsymbol{t} \in \mathbb{R}^{3\times1}$ are the rotation matrix and the translation vector between the camera and the target frames. Then, the reprojection error, defined by

$$\sum_i d(\boldsymbol{P}\boldsymbol{L}_i, \boldsymbol{l}_i), \tag{12}$$

is used as a measure of tracking error. In (12) $d(\cdot,\cdot)$ is a distance measure between each predicted ($\boldsymbol{P}\boldsymbol{L}_i$) and detected ($\boldsymbol{l}_i$) feature point on an image plane, while $\boldsymbol{L}_i \in \mathbb{R}^4$ and $\boldsymbol{l}_i \in \mathbb{R}^3$ are representing the 3D coordinates of a known fiducial marker on a target frame and the pixel coordinates of the detected feature point in the image plane, respectively, in homogeneous form.

As the projection matrix error is mainly attributed to the drifted tracking feature points, the reprojection error captures the feature tracker drift. Hence, the reprojection error is periodically evaluated, and if it drops below a pre-defined threshold value, feature tracking terminates and new feature points need to be matched and tracked. This prevents long time tracking drift, and keeps the pose determination accuracy bounded. Good feature point tracking directly allows accurate pose determination in the next step.

IV.D. Pose Determination

In the literature, there exist many n-point algorithms to iteratively or non-iteratively solve the absolute or relative pose estimation problem with respect to 3D target points. Each method has its own advantages and disadvantages. Nonlinear iterative methods suffer from getting trapped at a local minimum, depending on the initial guess, whereas a closed-form solution may be vulnerable to noisy data. A standard nonlinear iterative Levenberg-Marquardt algorithm is known to possibly converge to a local minimum. Meanwhile, Haralick's nonlinear iteration method [37] is known to be globally convergent at an infinite number of iterations, but errors will exist by limiting the maximum number of iterations, as it is done in practice.

In this paper the Efficient Perspective n-Point (EPnP) with $n = 4$ has been used to obtain raw pose estimation data. The Efficient Perspective n-Point (EPnP) algorithm [26] non-iteratively determines the camera pose using a linear approximation setting, given the intrinsic camera matrix and a set of n correspondences between a set of 3D points and their 2D projections (3D-2D correspondence) on the camera image plane. The computational complexity increases linearly with the number of correspondences, i.e., it is of order $O(n)$. The approach is applicable for $n \geq 4$, and it is shown that if all 3D points are on the same plane, a closed-form EPnP provides an accurate enough solution. It is common to refine the general PnP solution, by running an optimization step (reprojection minimization), but in this special case (co-planar EP4P), the EPnP eliminates the need for the refinement process.

V. Linear and Angular Velocity Estimation via a Probabilistic Graphical Model

The pose estimator processes images alone, and it does not directly provide velocity information, which has to be obtained by other means. In general, velocity is estimated using a high-pass filter, or a recursive filter (EKF, UKF, PF) smoothed by a dynamic model of the system. However, a high-pass filter yields a non-optimal approximate solution and may be corrupted by noise, whereas a recursive filter may not be accurate enough, since its state history is only encoded by a single Gaussian density (mean and covariance), so an unfortunate linearization choice cannot be corrected [38].

In this paper we have adopted a graphical model-based smoothing framework, which is known to provide better accuracy for state estimation by allowing incorrect linearization points, but by keeping (part of) the history of the state. A dynamic model of the process is used to predict and initialize the next state, and this builds a constraint between the states as shown in the graphical model in Fig. 3. The GTSAM (Georgia Tech Smoothing and Mapping) is a free smoothing and mapping library [39] developed for bundle adjustment in SLAM/SfM, and is known to have increased speed and accuracy. It uses a factor graph and variable ordering to improve numerical efficiency. In this paper, GTSAM has been used to implement a fixed-lag smoothing over the collected data. A brief outline of the method is given in the next section. More details can be found in Refs. [30, 31].

V.A. Kinematic and Measurement Models

In the model of the system, the state $\boldsymbol{x}$ consists of the relative pose and the relative linear and angular velocities. The control vector $\boldsymbol{u}$ includes the force and torque generated by thrusters and/or momentum exchange devices on-board the spacecraft. The camera measurements vector $\boldsymbol{z}$ consists of the pixel locations of the detected feature points. By assuming Gaussian process and measurement models, the discrete representation of the process model is given in the form

$$\boldsymbol{x}(k+1) = g_k(\boldsymbol{x}(k), \boldsymbol{u}(k)) + \boldsymbol{w}(k). \tag{13}$$

This equation describes the time evolution of the state $\boldsymbol{x}(k)$, where $\boldsymbol{w}(k)$ is a normally distributed zero-mean process noise with covariance matrix $\boldsymbol{\Lambda}_k$. The process noise represents the dynamics modeling error. We use a simple linear model (with $\boldsymbol{u}(k) = 0$) for the translation and rotation dynamics as follows

$$\boldsymbol{x}(k+1) = \Phi_k \boldsymbol{x}(k) + \Gamma_k \boldsymbol{w}(k), \tag{14}$$

where $\boldsymbol{x}(k) = [\boldsymbol{r}(k)\ \boldsymbol{v}(k)\ \boldsymbol{q}(k)\ \boldsymbol{\omega}(k)]^T$ is the state vector, composed of $\boldsymbol{r}$, the relative position between camera and target, $\boldsymbol{v}$, the relative linear velocity, $\boldsymbol{q}$, the quaternion describing the relative orientation of the camera frame and $\boldsymbol{\omega}$, the corresponding camera angular velocity. In (14) the matrix Φ_k is given by [40]

$$\Phi_k = \mathrm{e}^{A\Delta t} \approx \begin{bmatrix} I_{3\times3} & I_{3\times3}\Delta t & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{4\times3} & 0_{4\times3} & \Omega(\boldsymbol{\omega}(k)) & 0_{4\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3} \end{bmatrix}, \tag{15}$$

where

$$\Omega(\boldsymbol{\omega}(k)) = \begin{bmatrix} \cos\left(\frac{1}{2}\|\boldsymbol{\omega}(k)\Delta t\|\right) I_{3\times3} - [\boldsymbol{\eta}(k)\times] & \boldsymbol{\eta}(k) \\ -\boldsymbol{\eta}(k)^T & \cos\left(\frac{1}{2}\|\boldsymbol{\omega}(k)\Delta t\|\right) \end{bmatrix}, \qquad \boldsymbol{\eta}(k) = \frac{\boldsymbol{\omega}(k)}{\|\boldsymbol{\omega}(k)\|} \sin\left(\frac{1}{2}\|\boldsymbol{\omega}(k)\Delta t\|\right) \tag{16}$$

where

$$A = \begin{bmatrix} 0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{4\times3} & 0_{4\times3} & \Xi(\boldsymbol{\omega}) & 0_{4\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix}, \qquad \Xi(\boldsymbol{\omega}) = \frac{1}{2}\begin{bmatrix} -[\boldsymbol{\omega}\times] & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix}, \tag{17}$$

and Γ_k is computed from

$$\Gamma_k = \left(\int_0^{\Delta t} \mathrm{e}^{A\tau}\, \mathrm{d}\tau\right) G \approx \begin{bmatrix} \frac{1}{2m} I_{3\times3}\Delta t^2 & 0_{3\times3} \\ \frac{1}{m} I_{3\times3}\Delta t & 0_{3\times3} \\ 0_{4\times3} & 0_{4\times3} \\ 0_{3\times3} & \boldsymbol{J}^{-1}\Delta t \end{bmatrix}, \qquad G = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} \\ \frac{1}{m} I_{3\times3} & 0_{3\times3} \\ 0_{4\times3} & 0_{4\times3} \\ 0_{3\times3} & \boldsymbol{J}^{-1} \end{bmatrix}, \tag{18}$$

with m is the mass of the chaser spacecraft and $\boldsymbol{J}$ is its inertia matrix.

The discrete measurement model is given by

$$\boldsymbol{z}(k) = h_k(\boldsymbol{x}(k)) + \boldsymbol{n}(k), \tag{19}$$

which represents a pin-hole camera perspective projection model, where $\boldsymbol{n}(k)$ is normally distributed zero-mean measurement noise with covariance matrix $\boldsymbol{Q}_k$. The pin-hole camera measurement equation is given by the perspective projection matrix equation

$$\boldsymbol{p} = \boldsymbol{K}[\boldsymbol{R}\ \ \boldsymbol{t}]\boldsymbol{L} \tag{20}$$

with $\boldsymbol{p}$ and $\boldsymbol{L}$ representing the predicted pixel locations in the image frame and the 3D coordinates of the known fiducial markers on a target frame, respectively, in homogeneous form. The rotation matrix $\boldsymbol{R}$ and the translation vector $\boldsymbol{t}$ represent the transformation between the camera and the target frames, calculated

from the previous pose estimate and are functions of the relative pose between the chaser spacecraft and the target, that is, $\boldsymbol{R} = \boldsymbol{R}(\boldsymbol{r}, \boldsymbol{q})$ and $\boldsymbol{t} = \boldsymbol{t}(\boldsymbol{r}, \boldsymbol{q})$.

In our problem $\boldsymbol{z} = [\boldsymbol{p}_1^T \ \boldsymbol{p}_2^T \ \boldsymbol{p}_3^T \ \boldsymbol{p}_4^T]^T$ since the target is composed of four distinct markers (see Fig. 5). The details of the derivation of the exact expression h_k in (19) involves simple geometric arguments and is omitted for the sake of brevity.

V.B. Factor Graph

Instead of representing the navigational smoothing problem in conventional matrix form, we have used a probabilistic graphical model, namely, a factor graph. A factor graph [41] is a bipartite graph $G = (\mathcal{F}, \Theta, \mathcal{E})$ consisting of factor nodes $f_i \in \mathcal{F}$, variable nodes $\theta_j \in \Theta$, and edges $e_{ij} \in \mathcal{E}$. The factor graph G defines the factorization of a function $f(\Theta)$ according to the formula

$$f(\Theta) = \prod_i f_i(\Theta_i), \tag{21}$$

where Θ_i is the set of variables θ_j connected by an edge e_{ij} to factor f_i. Each factor represents an error function between predicted and actual measurement, written as

$$f_i(\Theta_i) \triangleq d(x_i - g_i(\Theta)) = \boldsymbol{c_i}, \tag{22}$$

or

$$f_j(\Theta_j) \triangleq d(z_j - h_j(\Theta)) = \boldsymbol{m_j}, \tag{23}$$

where $d(\cdot)$ denotes a distance measure or a cost function. For instance, in case of Gaussian noise distributions $d(\cdot)$ is the squared Mahalanobis distance.

The monocular camera-based relative navigation problem can be represented in terms of a factor graph as shown in Fig. 3.

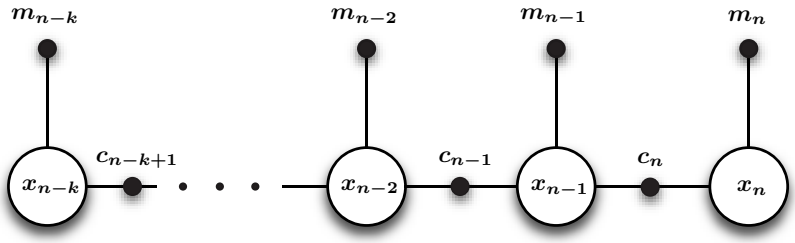


Figure 3. Factor graph representation for fixed-lag smoothing with a k sliding window width. Filtering corresponds to the case when only the most recent state x_n is kept. There are two kinds of factors: unary camera measurement factors m and binary control factors c relating the state nodes.

The optimal estimate $\hat{\Theta}$ is obtained via nonlinear least squares optimization,

$$\hat{\Theta} = \arg\min_{\Theta} f(\Theta) = \arg\min_{\Theta}[-\log f(\Theta)], \tag{24}$$

assuming a Gaussian noise model. Conventionally, smoothing is the process optimizing all variables using all measurements. If only the last few states are kept in the graph by marginalizing out the previous states, it is called fixed-lag smoothing. By the same token, when only the most recent state is kept, the process is called filtering.

Since the fiducial marker configuration ($\boldsymbol{L}$) and the calibration matrix ($\boldsymbol{K}$) are known a priori, this measurement only depends on the current state $\boldsymbol{x}$, which defines a unary factor $\boldsymbol{m}$ on the variable node $\boldsymbol{x}$. On the other hand, the state evolution depends on the previous and current states, so a binary factor $\boldsymbol{c}$ exists between the variable nodes. Figure 3 illustrates the factor graph model used in our framework that uses a monocular camera.

Note that factor graph-based smoothing is just an alternative estimation method to filtering. Since the prior (part of the) history of all measurements is kept, it typically provides better accuracy and robustness than filtering, albeit at the expense of longer computational times. The SLAM-based smoothing technique used in this work (GTSAM), however, enables real-time implementation for vision-based relative navigation state estimation, as it was demonstrated experimentally in the next section.

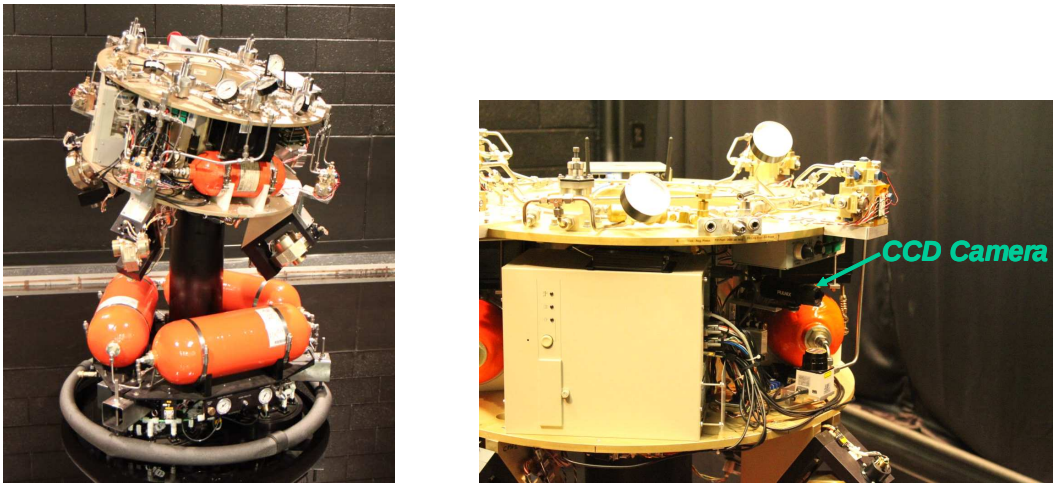
VI. Experimental Setup and Results

VI.A. Spacecraft Simulator Test Bed

The proposed algorithm has been tested on the 5-DOF spacecraft simulator facility for Autonomous Spacecraft Testing of Robotic Operations in Space (ASTROS), located at the Dynamics and Control Systems Laboratory at the School of Aerospace Engineering at Georgia Tech. The Georgia Tech ASTROS platform, shown in Fig. 4, involves two stages, the lower stage (the pedestal) and the upper stage (main spacecraft bus) moving over an extremely flat epoxy floor, of dimensions approximately 14 ft x 14 ft, thus simulating almost friction-free conditions.

The lower stage consists of four high-pressure air storage vessels, three linear air-bearing pads, a hemi-spherical air-bearing cup (connecting the lower and upper stages), along with the dedicated electronics and power supply. The three linear air pads allow the lower stage to float on a very thin (air gap about 70-80 microns) cushion of air. The main structure of the ASTROS is the upper stage, whose main operational characteristics can be found in Ref. [42]. The upper stage represents a typical spacecraft "bus" and is made of a two-level brass structure that is supported on a hemi-spherical air bearing, allowing rotation of the upper stage with respect to the supporting pedestal about all three axes (± 30 deg about the x and y axes and a full rotation about the z axis). A set of 12 thrusters and a set of four variable-speed control moment gyros (VSCMGs) can be used to apply external forces and moments on the platform. Several sensors (rate gyros, accelerometers, magnetometer, IMU) provide attitude and attitude rate information. Most relevant to this work, a CCD camera (TMS-730p by Pulnix) is used for vision-based relative navigation. An external 2D laser scanner (LMS-200 from SICK) provides 1- 1.5 (cm) accuracy for inertial position information. More accurate absolute position and attitude information is provided by a 6-camera VICON system.

The xPC Target environment (with Embedded Option) from Mathworks, along with Simulink models and the RealTime Workshop, is used for real-time controller implementation on board the platform.



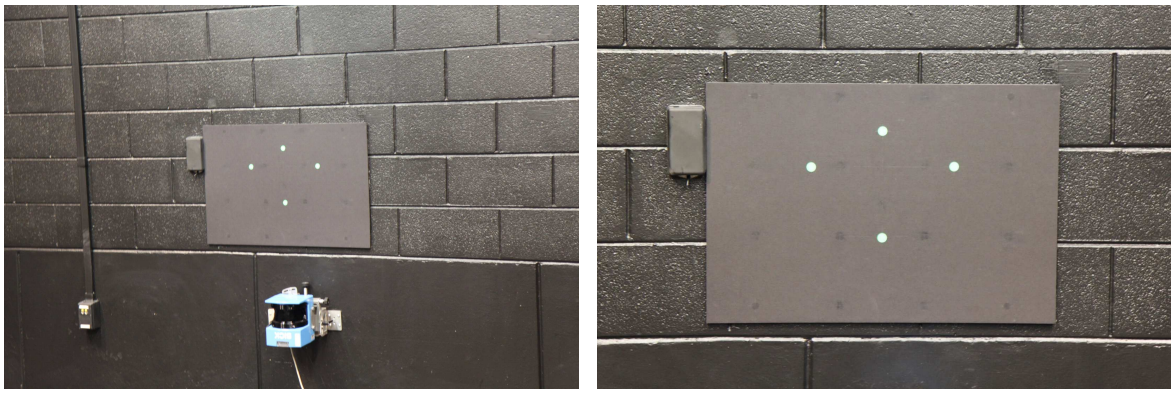
(a) 5 DOF Spacecraft Simulator (b) On-board CCD Camera

Figure 4. Experimental platform.

VI.B. Target

Several target designs have been proposed and tested in the literature. Matching of at least four coplanar or six free-in-space fiducial markers with a known configuration are required to estimate full relative pose (6 DOF) with respect to a target without ambiguities [23]. Several approaches have been used in the literature to uniquely identify the fiducial markers: beacons or LEDs at different blinking frequencies, concentric circles with different area ratios, reflectors in a unique configuration, circular markers, etc. See [11] and [12] for nice overviews of the current state of the art.

For our experiments we have used a planar target with four circular markers. Each centroid of the circular markers is used as a feature point, and the approximate location of the centroid can be detected by a corner detector when the spacecraft is further away from the target. At close range, the exact centroid location can be found by using a scale invariant blob detector. In addition, the circular fiducial markers emulate an image of an infra-red LED array, which is often used for robust feature detection under changing illumination conditions. Hence the same feature point detection algorithm can be applied in such a case as well.



(a) Target on the Wall (b) Fiducial markers on the Target

Figure 5. Target used in the experiments.

A minimal number of feature points to be matched is required for robust feature point matching. Redundant feature points can be used to increase the robustness owing to potential loss of some fiducial markers. Instead of matching each feature point, in this paper, feature-pattern matching based on GMM point-set registration is proposed to increase the robustness of matching between an a priori known feature point configuration and a set of detected feature points in the image. It is shown that robust target acquisition is possible even when using the minimal allowable number (four) of fiducial markers.

VII. Experimental Results

The proposed relative navigation algorithm was evaluated by conducting a series of feature detection and selection tests, along and target tracking and re-acquisition experiments.

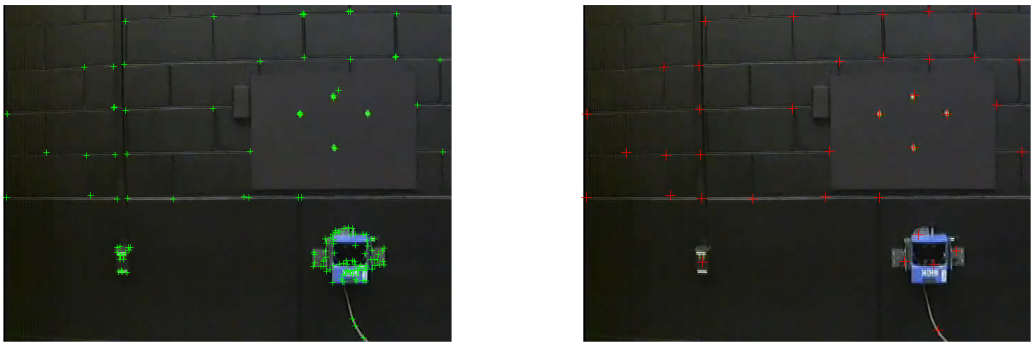
VII.A. Feature Detection and Selection/Filtering Test

This section discusses the test of the feature detection and selection/filtering step on a video captured with the CCD camera on the experimental platform. The marker pattern is mounted on one of the walls as described in Section VI.B. Moreover, to test the robustness of the algorithm, some clustered objects are placed in the environment. The video is captured by smoothly moving the camera in front of the marker, then rotating the camera horizontally to simulate relative orbiting. While the camera is rotating, the pattern gradually moves out of the view of the camera. At the end of the video, the camera rotates back to the view towards the pattern in order to test the re-acquisition of the pattern. The resolution of the video is 640×480 and the frame-rate is 18.0 Hz.

The video is processed every five frames. In the first step, we use the scheme described in Section IV.B to detect a raw set of visual features. We then follow the three sub-steps in Section IV.B. In the neighborhood

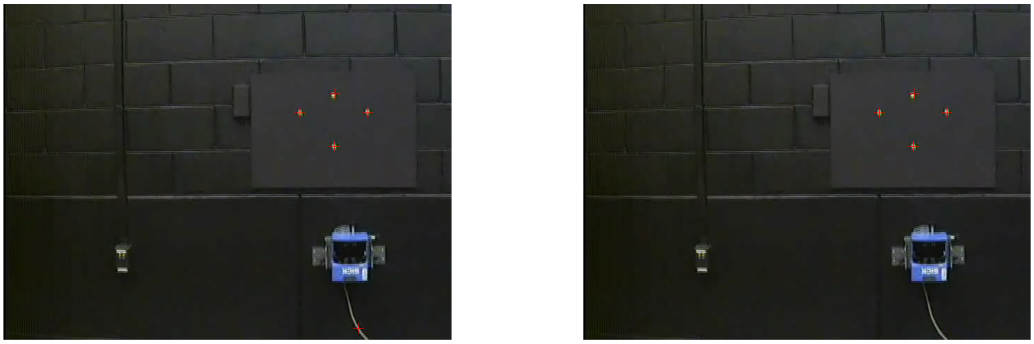
filtering step, the radius of the neighborhood is 10-pixel; in the color-based filtering step, the local block is of size 10×10 pixels, with $K = 25$, $\tau = 1.315$, $\delta = 0.2$; in the coherence filtering step, R_M is set as 40 pixels. These parameters were optimized empirically.

The code was implemented in MATLAB 2013b with the Computer Vision Toolbox. The experiment was conducted on a 64-bit Linux machine with a 2.40 Hz CPU core with 6Gb RAM. The average processing time of feature detection and selection/filtering is 0.4524 seconds. Typical results are illustrated in Fig. 6, in which each sub-step selects out the features on the marker gradually and in the final selected set only the features on the marker are kept. Figures 7 and 8 show two other typical frames from the same video. In Fig. 7, when the camera rotates, it captures a clustered view without the marker. The feature detector detects multiple features (shown in Fig. 7(a)), but after the selection/filtering step all these features are removed (shown in Fig. 7(b)). Figure 8 illustrates the result when the camera re-acquires a view with the marker after the target has been lost for some time. In this case, the selection/filtering step also effectively selects the correct feature set. It is interesting to note that even though the processed video frames are of poor quality (aliasing owing to the CCD camera motion is clearly evident in Figs. 7 and 8), the algorithm works very well. More accurate results are expected by replacing the current CCD camera on-board the experimental platform.



(a) Originally detected features.

(b) Features after neighborhood filtering.



(c) Features after neighborhood filtering and color-based filtering.

(d) Features after neighborhood filtering,color-based filtering and coherence filtering.

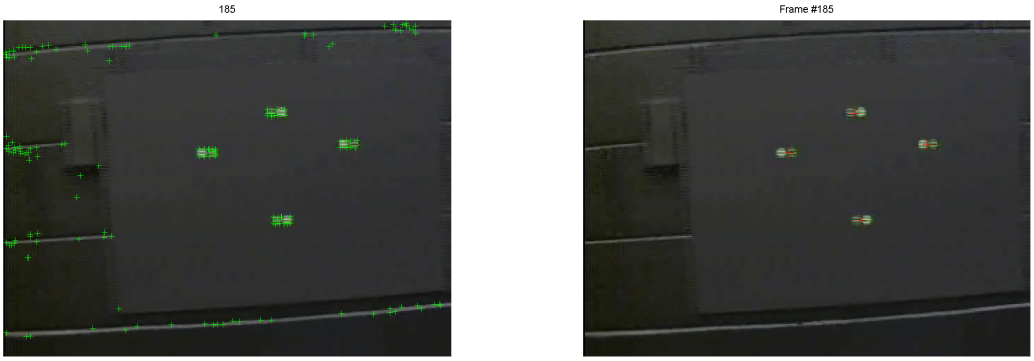
Figure 6. Feature selection/filtering result of a typical frame. After the selection/filtering step, features on the marker are effectively selected.

VII.B. Target Re-acquisition Test

The results from Section VII.A were further used to test the feature-pattern matching algorithm discussed in Section IV.B. The frames with more than two selected features were tested. In the test, the variance of each Gaussian component in GMMs is set to 0.5, and the initial parameters of the affine transformation, i.e., the rotation, translation and scale, were set randomly. The nonlinear optimization process employs the

(a) Originally detected features. (b) Features after the selection/filtering step.

Figure 7. Feature selection/filtering result of a typical frame with cluttered objects but without the marker.



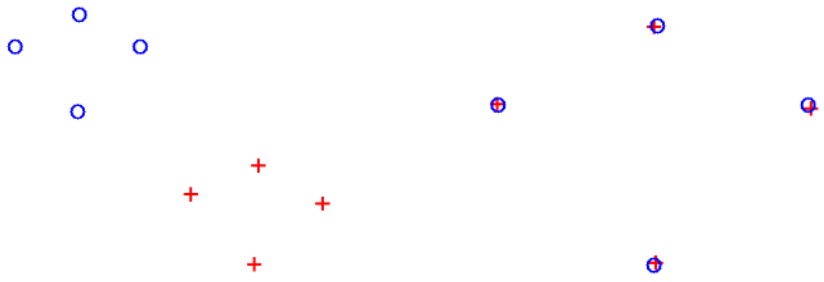
(a) Originally detected features. (b) Features after the selection/filtering step.

Figure 8. Feature selection/filtering result when the camera is back to viewing the marker.

Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) method [43]. The average processing time for each frame is 0.0619 seconds.

Two typical results are shown in Figures 9 and 10, when the camera is far from and close to the marker, respectively. It can be observed that in both cases the algorithm gives accurate results. Figure 11 further illustrates a situation when the cardinality of the feature-set and the pattern are not the same. In Figure 11, one less point is observed in the feature-set than the pattern, but the matching algorithm still gives the correct matching, which demonstrates the robustness of our feature-pattern matching algorithm.

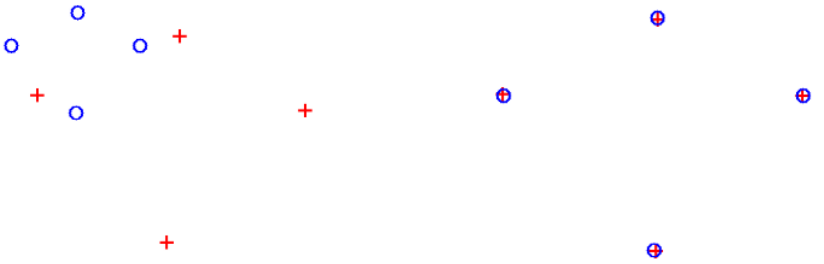
Besides the point-set registration step in the target re-acquisition algorithm of this paper, there are many other point-set registration approaches that can be used for pattern re-acquisition. Some state-of-the-art registration algorithms include [20], [18], [19]. The data in this experiment was further tested with these three methods and the processing time compared with our method is shown in Table 1. All of the algorithms are very fast and process the video in real-time. The algorithm in this paper is able to recover the affine transformation directly, while algorithms [20], [18], [19] all use bases/spline functions to model the transformation (Ref. [20] uses thin-plate splines and [18,19] use the Gaussian Radial Basis Functions), which does not allow them to recover the affine transformation directly. Since the pattern re-acquisition results will be further used in the tracking process, the affine transformation gives significant prior knowledge of the feature poses. This justifies the choice of the algorithm discussed in Section IV.B for pattern re-acquisition for our problem.



(a) Original point-sets (blue-circles: pattern; red-crosses: observed features)

(b) Matched point-sets (blue-circles: pattern; red-crosses: observed features after recovered affine transformation)

Figure 9. Feature-pattern matching result when the camera is far away from the marker.



(a) Originally detected features.

(b) Features after the selection/filtering step.

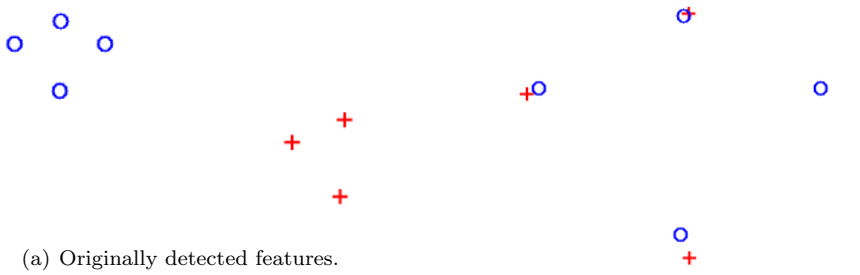
Figure 10. Feature-pattern matching result when the camera is close to the marker. This is from the frame when the camera re-captures the marker in the latter part of the video.

Table 1. Comparison of processing time of different algorithms on the tested dataset.

Methods	Proposed method	Ref. [20]	Ref. [18]	Ref. [19]
Time/Frame (sec.)	0.0619	0.0603	0.0628	0.0619

VIII. Conclusions

This paper proposes a new relative navigation algorithm based of feature detection and tracking with a known target, designed for spacecraft rendezvous and proximity operations in space. The algorithm can operate over a wide range of distances from a target, it has automated re-acquisition capability, and



(a) Originally detected features.

(b) Features after the selection/filtering step.

Figure 11. Feature-pattern matching result when the cardinalities of the feature-set and the pattern are not the same.

provides better estimation accuracy and better robustness than EKF-based methods. Since the algorithm is based on a graphical model-based SLAM/SfM framework, it can easily accommodate additional sensors (via the incorporation of the additional associated factors), fuse multi-rate sensor measurements, and it can be adapted to cases of relative navigation with respect to an uncooperative target. These extensions are currently under investigation and the results will be reported in the near future.

References

[1]Gaylor, D. and Lightsey, E. G., "GPS/INS Kalman filter design for spacecraft operating in the proximity of the international space station," *Proc. of AIAA Guidance, Navigation, and Control Conference and Exhibit*, Addison-Wesley, 2003.

[2]Kasai, T., Oda, M., and Suzuki, T., "Results of the ETS-7 Mission-Rendezvous docking and space robotics experiments," *Artificial Intelligence, Robotics and Automation in Space*, Vol. 440, 1999, p. 299.

[3]How, J., Twiggs, R., Weidow, D., Hartman, K., and Bauer, F., "Orion: A low-cost demonstration of formation flying in space using GPS," *Proceedings of AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 1998, pp. 10–12.

[4]DiMatteo, J., Florakis, D., Weichbrod, A., and Milam, M., "Proximity operations testing with a rotating and translating resident space object," *AIAA Guidance, Navigation, and Control Conference*, Chicago, IL, 2009.

[5]Ruel, S. and Luu, T., "STS-128 on-orbit demonstration of the TriDAR targetless rendezvous and docking sensor," *IEEE Aerospace Conference*, 2010, pp. 1–7.

[6]Nolet, S., Kong, E., and Miller, D., "Autonomous Docking Algorithm Development and Experimentation Using the SPHERES Testbed," *Proceedings of SPIE*, Vol. 5419 of *Spacecraft Platforms and Infrastructure*, August 2004, pp. 1–15.

[7]Walker, L. and Spencer, D., "Automated Proximity Operations Using Image-Based Relative Navigation," *26th Annual AIAA/USU Conference on Small Satellites*, Logan, UT, August 2012.

[8]Shkurti, F., Rekleitis, I., and Dudek, G., "Feature Tracking Evaluation for Pose Estimation in Underwater Environments," *Computer and Robot Vision*, May 2011, pp. 160–167.

[9]Shi, J. and Tomasi, C., "Good features to track," *Computer Vision and Pattern Recognition*, June 1994, pp. 593–600.

[10]Bay, H., Tuytelaars, T., Van Gool, L., and Van Gool, L., "SURF: Speeded Up Robust Features." *Computer Vision-ECCV 2006*, Vol. 3951, No. 3, 2006, pp. 404–417.

[11]Augenstein, S., *Monocular Pose and Shape Estimation of Moving Targets for Autonomous Rendezvous and Docking*, Ph.D. thesis, Dept. of Aeronautics and Astronautics, Stanford University, Palo Alto, CA, June 2011.

[12]Flewelling, B., *3d Multi-field Multi-scale Features From Range Data In Spacecraft Proximity Operations*, Ph.D. thesis, Texas A&M University, College Station, TX, 2012.

[13]Terui, F., Kamimura, H., and Nishida, S., "Motion estimation to a failed satellite on orbit using stereo vision and 3D model matching," *9th International Conference on Control, Automation, Robotics and Vision*, IEEE, 2006, pp. 1–8.

[14]Kelsey, J. M., Byrne, J., Cosgrove, M., Seereeram, S., and Mehra, R. K., "Vision-based relative pose estimation for autonomous rendezvous and docking," *IEEE Aerospace Conference*, IEEE, 2006, pp. 20–40.

[15]Tweddle, B. E., "Relative Computer Vision based Navigation for Small Inspection Spacecraft," *Proc. AIAA Guidance, Navigation and Control Conference*, Portland, OR, 2011.

[16]Lowe, D. G., "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, Vol. 60, No. 2, 2004, pp. 91–110.

[17]Kolesov, I., Lee, J., Vela, P., and Tannenbaum, A., "A stochastic approach for non-rigid image registration," *IS&T/SPIE Electronic Imaging*, International Society for Optics and Photonics, 2013, pp. 86550U–86550U.

[18]Myronenko, A., Song, X., and Carreira-Perpinán, M. A., "Non-rigid point set registration: Coherent point drift," *Advances in Neural Information Processing Systems*, 2006, pp. 1009–1016.

[19]Tsin, Y. and Kanade, T., "A Correlation-Based Approach to Robust Point Set Registration," *In ECCV*, 2004.

[20]Chui, H. and Rangarajan, A., "A new point matching algorithm for non-rigid registration," *Computer Vision and Image Understanding*, Vol. 89, No. 2, 2003, pp. 114–141.

[21]Jian, B. and Vemuri, B. C., "A robust algorithm for point set registration using mixture of Gaussians," *Computer Vision, 2005. Tenth IEEE International Conference on*, Vol. 2, IEEE, 2005, pp. 1246–1251.

[22]Dhome, M., Richetin, M., Lapreste, J., and Rives, G., "Determination of the Attitude of 3D Objects from a Single Perspective View," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 11, No. 12, 1989, pp. 1265–1278.

[23]Fischler, M. A. and Bolles, R. C., "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography." *Communications of the ACM*, Vol. 24, No. 6, 1981, pp. 381–395.

[24]Gao, X.-S., Hou, X.-R., Tang, J., and Cheng, H.-F., "Complete Solution Classification for the Perspective-Three-Point Problem," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 25, No. 8, 2003, pp. 930–943.

[25]Quan, L. and Lan, Z., "Linear N-Point Camera Pose Determination," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 21, No. 8, 1999, pp. 774–780.

[26]Lepetit, V., Moreno-Noguer, F., and Fua, P., "EPnP: An Accurate O(n) Solution to the PnP Problem," *Int. J. Comput. Vision*, Vol. 81, No. 2, 2009, pp. 155–166.

[27]Kaess, M., Williams, S., Indelman, V., Roberts, R., Leonard, J., and Dellaert, F., "Concurrent Filtering and Smoothing," *Intl. Conf. on Information Fusion*, Singapore, July 2012, pp. 1300–1307.

[28]Koller, D. and Friedman, N., *Probabilistic Graphical Models: Principles and Techniques*, MIT Press, 2009.

[29]Thrun, S., Burgard, W., and Fox, D., *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*, MIT Press, 2005.

[30]Kaess, M and Ranganathan, A and Dellaert, F, "iSAM: Incremental Smoothing and Mapping," *IEEE Transactions on Robotics*, Vol. 24, No. 6, 2008, pp. 1365–1378.

[31]Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J., and Dellaert, F., "iSAM2: Incremental Smoothing and Mapping Using the Bayes Tree." *The International Journal of Robotics Research*, Vol. 31, No. 2, 2011, pp. 217–236.

[32]Indelman, V., Williams, S., Kaess, M., and Dellaert, F., "Factor Graph Based Incremental Smoothing in Inertial Navigation Systems," *Intl. Conf. on Information Fusion*, Singapore, July 2012, pp. 2154–2161.

[33]Harris, C. and Stephens, M., "A Combined Corner and Edge Detector," *Proceedings of The Fourth Alvey Vision Conference*, 1988, pp. 147–151.

[34]Jian, B. and Vemuri, B. C., "Robust point set registration using gaussian mixture models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 33, No. 8, 2011, pp. 1633–1645.

[35]Horn, R. A. and Johnson, C. R., *Matrix analysis*, Cambridge university press, 2012.

[36]Lucas, B. D. and Kanade, T., "An iterative image registration technique with an application to stereo vision," *International Joint Conference on Artificial Intelligence*, Vol. 81, 1981, pp. 674–679.

[37]Haralick, R., Joo, H., Lee, C., Zhuang, X., Vaidya, V., and Kim, M., "Pose Estimation from Corresponding Point Data." *IEEE Transaction on Systems Man and Cybernetics*, Vol. 19, No. 6, 1989, pp. 1426 – 1446.

[38]Dellaert, F. and Kaess, M., "Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing." *The International Journal of Robotics Research*, Vol. 25, No. 12, 2006, pp. 1181–1203.

[39]"GTSAM project in Borg Lab, Georgia Institute of Technology," `https://borg.cc.gatech.edu/projects/gtsam`, Accessed: 2013-07-28.

[40]Kim, S.-G., Crassidis, J. L., Cheng, Y., Fosbury, A. M., and Junkins, J. L., "Kalman Filtering for Relative Spacecraft Attitude and Position Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 1, 2007, pp. 133–143.

[41]Kschischang, F. R., Frey, B. J., and Loeliger, H.-A., "Factor Graphs and the Sum-Product Algorithm," *IEEE Transactions on Information Theory*, Vol. 47, 1998, pp. 498–519.

[42]Cho, D., Jung, D., and Tsiotras, P., "A 5-dof Experimental Platform for Spacecraft Rendezvous and Docking," *AIAA Infotech at Aerospace Conference*, April 6–9 2009, Seattle, WA.

[43]Liu, D. C. and Nocedal, J., "On the limited memory BFGS method for large scale optimization," *Mathematical programming*, Vol. 45, No. 1-3, 1989, pp. 503–528.